

*Session Code: ARC 420*

# *architecture & infrastructure*

## Transaction Support Under The Covers: Transactional NTFS, KTM, and WinLS

Rajeev Nagar

rajeevn@microsoft.com

Surendra Verma

sverma@microsoft.com

**PDC**<sup>03</sup>  
Dana Groff

dgroff@microsoft.com

Make the connection

**Microsoft**<sup>®</sup>

# WinFX Developer Preview

Tools

Visual Studio .net  
Visual Basic .net  
Visual C++ .net  
Visual C# .net  
Visual J# .net

## Client Application Model

### Avalon

System.Windows

### Windows Forms

System.Windows.Forms

## Web & Service Application Model

### ASP.NET / Indigo

System.Web

## Data Systems Application Model

### Win FS

System.Storage

### Yukon

System.Data.SqlServer

## Mobile PC & Devices Application Model

### Compact Framework

System.Windows.Forms

### Mobile PC Optimized

System.Windows

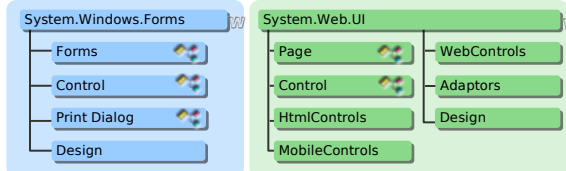
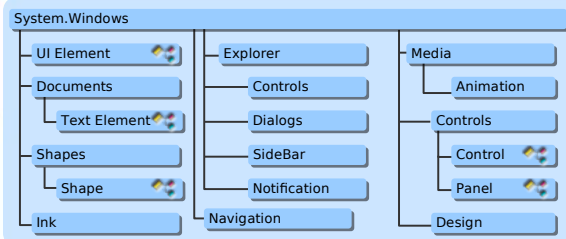
## Command Line

System.Console

### NT Service

System.ServiceProcess

## Presentation



System.Help

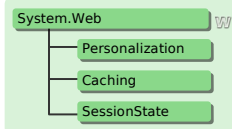
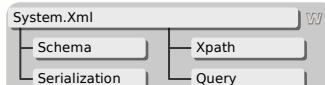
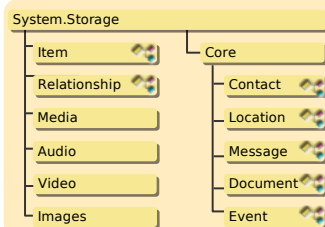
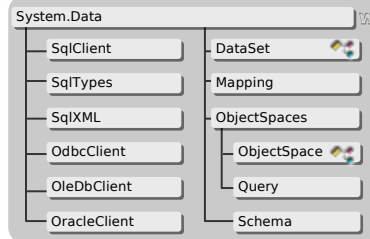
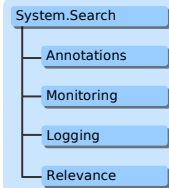
System.Drawing

System.NaturalLanguageServices

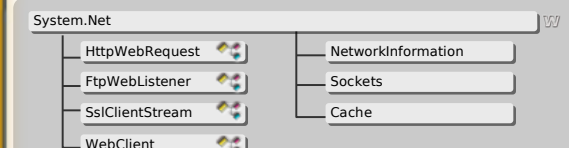
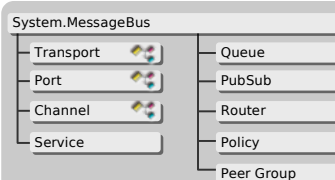
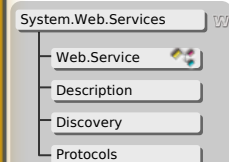
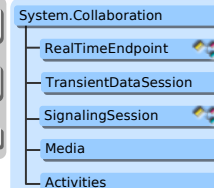
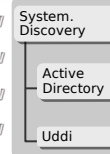
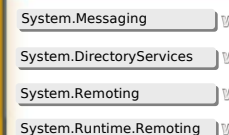
System.Speech

- Recognition
- Synthesis

## Data

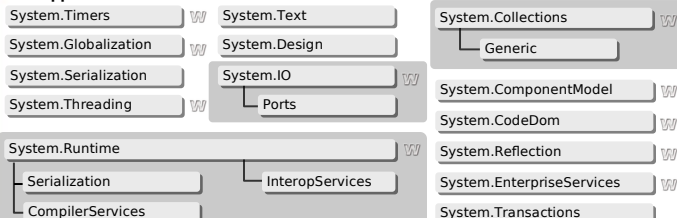


## Communication

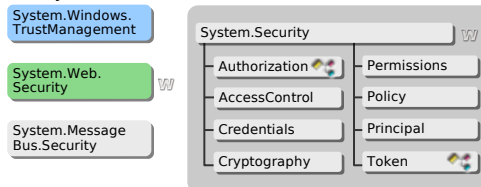


## Fundamentals

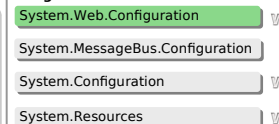
### Base & Application Services



### Security



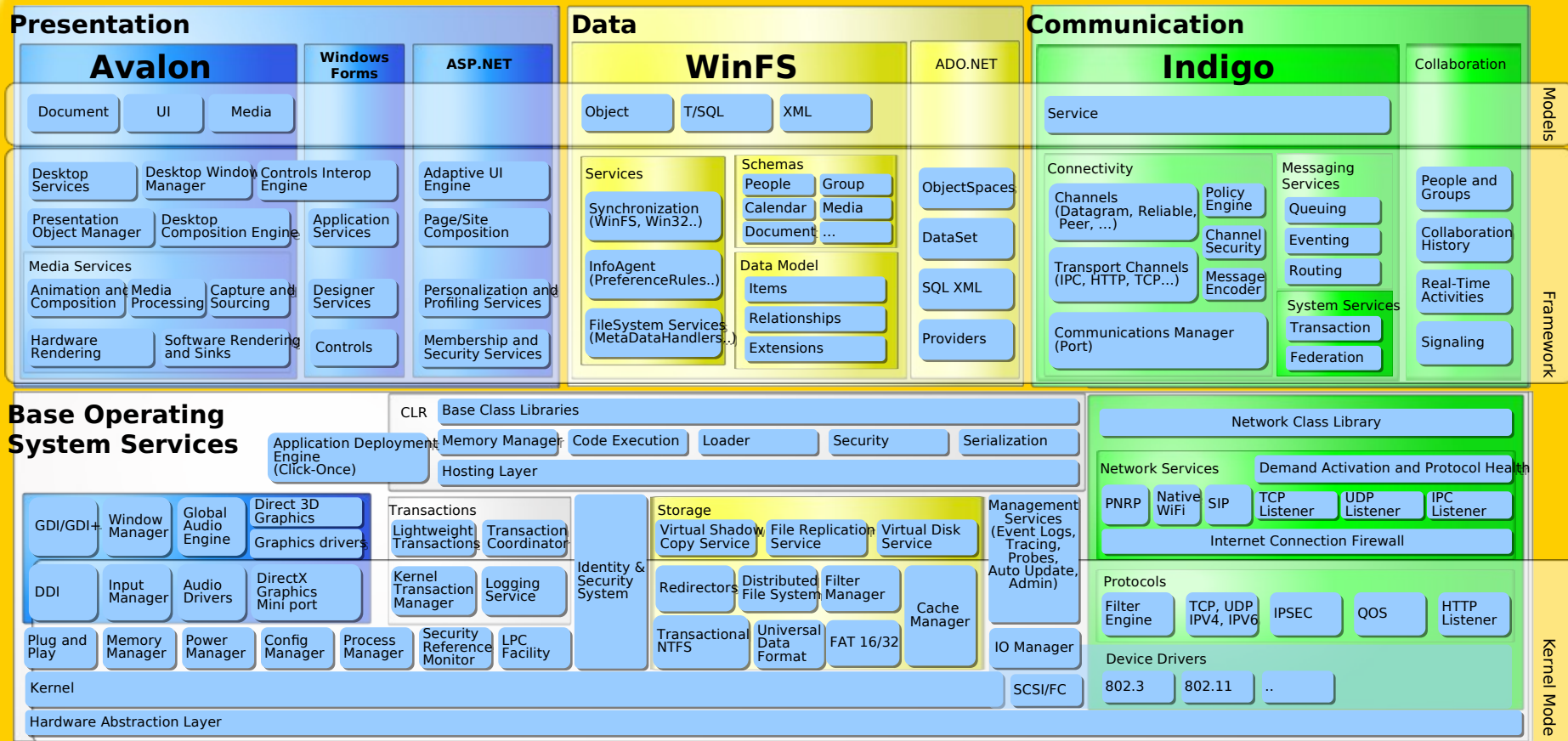
### Configuration



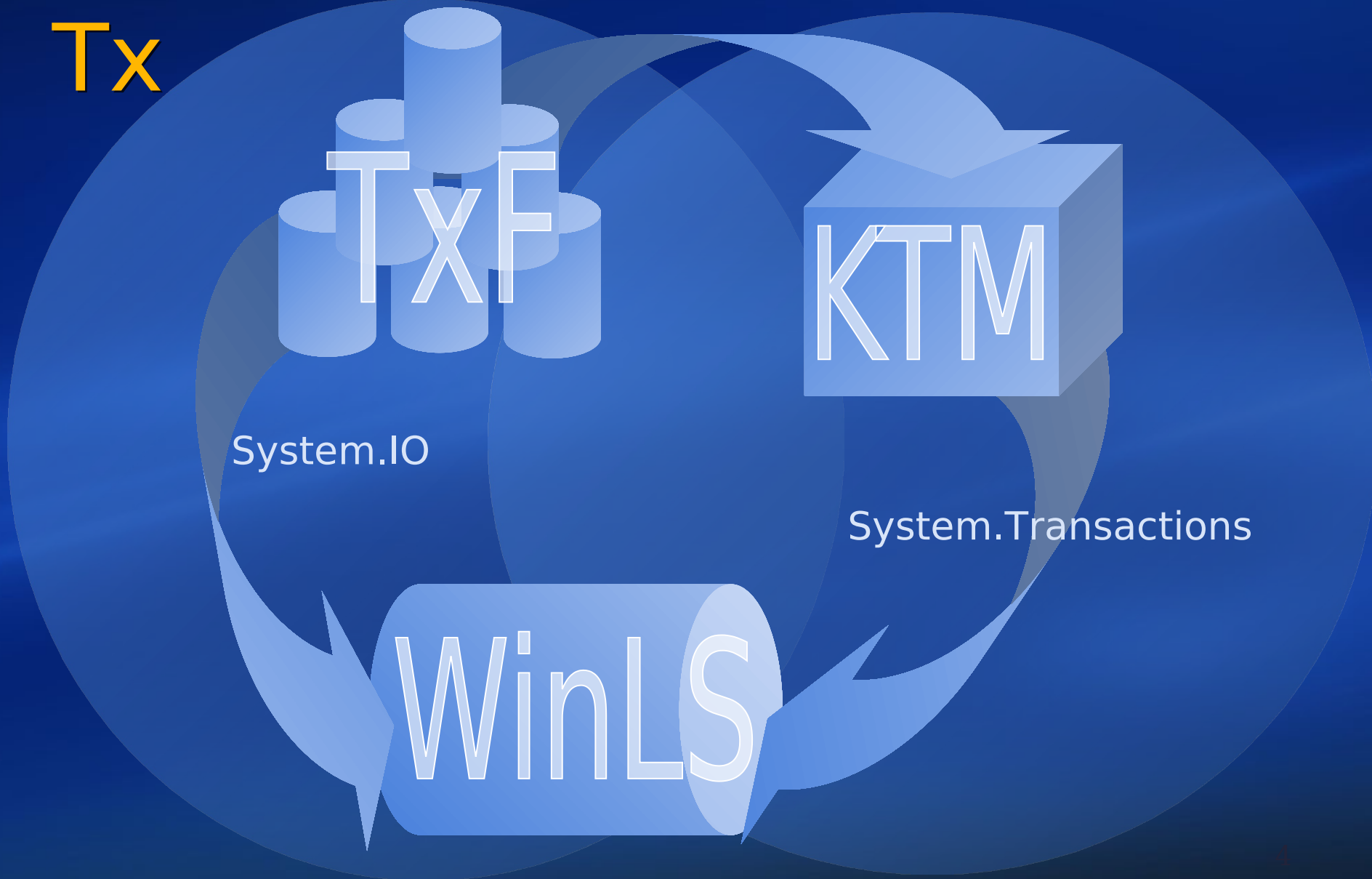
### Deployment/Management



# Longhorn Architecture



# Kernel Infrastructure for Tx



# Transactional NTFS (TxF)

Surendra Verma  
Software Design Engineer (Lead)  
Core File Services  
[sverma@microsoft.com](mailto:sverma@microsoft.com)



# Transactional NTFS (TxF)

## demo

Web site  
publishing

**PDC**<sup>03</sup>

Make the connection

# Functionality

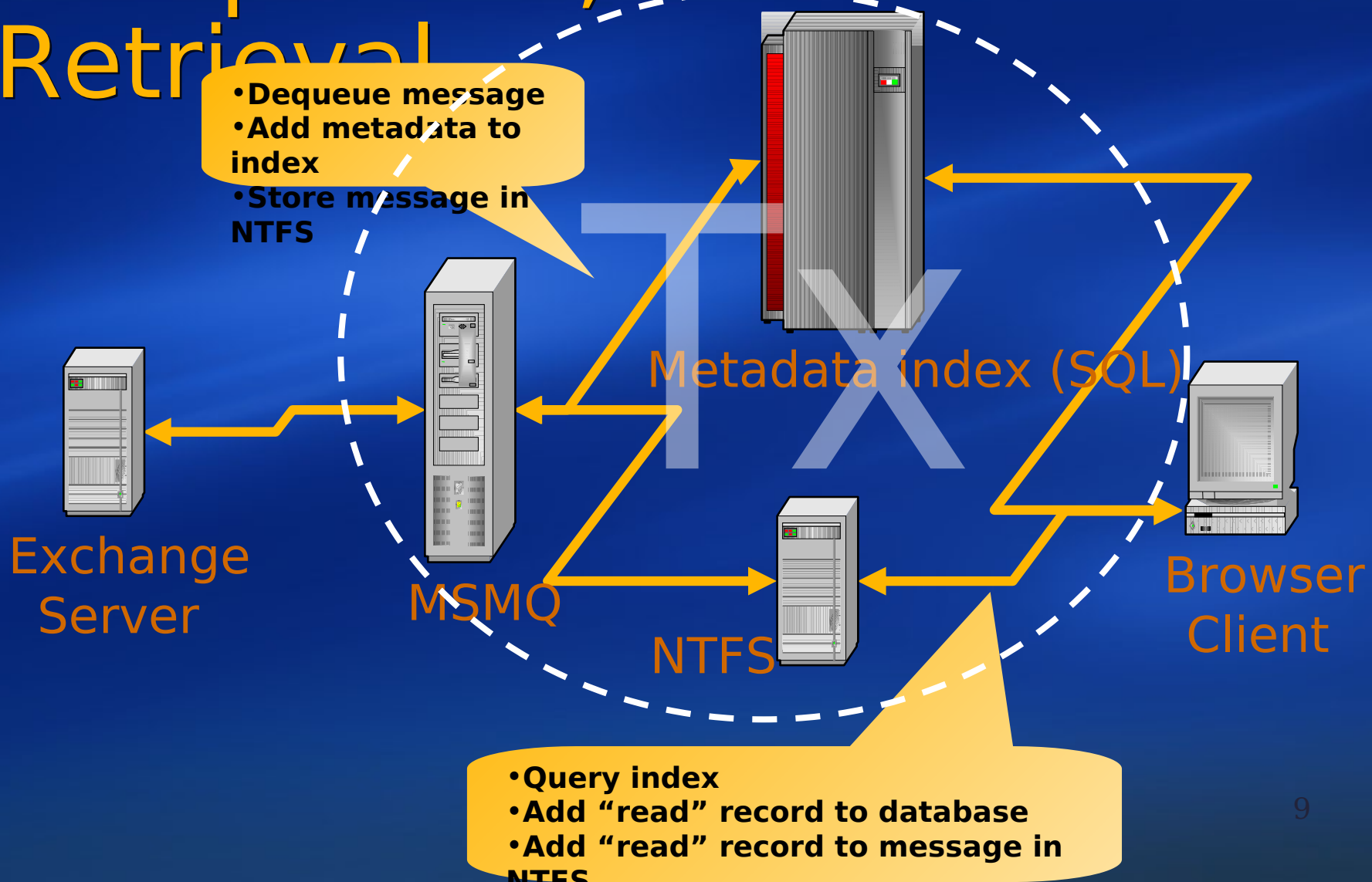
- Transaction Support for *all* NTFS file operations
  - Full Atomicity, Consistency, Isolation, Durability
  - Allow an arbitrary number of file system operations as an atomic unit
  - Reads, writes, file creations, deletions, renames, etc.
  - Memory mapping
- Network file access via SMB

# Usage Scenarios For TxF

- Safe and efficient in-place update of documents
- Transactional multi-file update and network transfer
- Files in NTFS, related data in another transactional store such as SQL
- WinFS Filestream (File Backed)



# Example Usage - Compliance, Archive & Retrieval



# Transactional NTFS (TxF)

## demo

## Transactional File I/O

**PDC**<sup>03</sup>

Make the connection

# Isolation And Locking

- Transactions don't see changes made by other transactions
- Isolation for non-transacted readers
- File is the unit of locking
- Readers don't block

# Logging Modes

- Simple Mode

- Only before-images of changes logged
- Minimal logging for common cases
  - Create a new file and write to it
  - Delete or rename a file
  - Extend a file
  - Fully truncate or overwrite a file

- Full Mode

- Full change history over time in log<sup>2</sup>

# Build Powerful Solutions

## Disaster recovery

- Specify and backup a set of files
- Set Logging Mode to Full
- Continuously Archive log
- Restore the set of files and archived log from backup
- Roll-forward using the log to a chosen point in time

# Resource Units (RU)

- Subset of a volume:
  - You create it for a directory tree
  - Keep data and transactional meta-data in one unit
  - Unit moves as a whole
  - You control start, stop and policies
- System-managed Resource Unit
  - Includes files not in any other RU
  - Self managed – logging mode, ...



# How To Program To It

- System.IO,  
System.Transaction, Win32
  - Start Transaction
  - Associate with current context
  - Do multiple file operations
  - Commit/Rollback Transaction
- Try Win32 APIs in Hands On Labs

# Kernel (mode) Transaction Manager

Rajeev Nagar  
Lead Program Manager  
Core File Services  
[rajeevn@microsoft.com](mailto:rajeevn@microsoft.com)

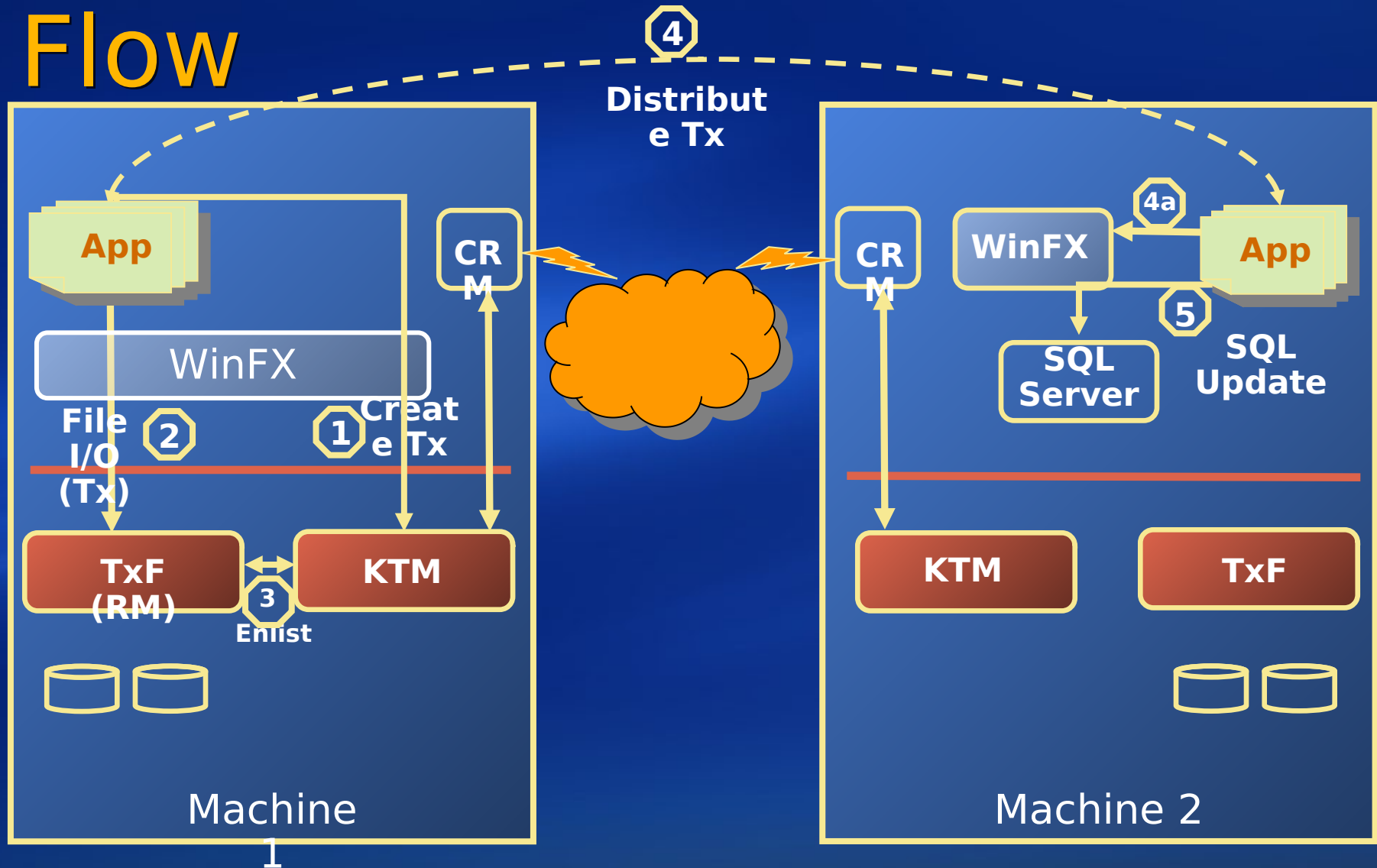
**PDC**<sup>03</sup>

Make the connection

# Technology Overview

- Kernel infrastructure for transaction support
  - Services for both kernel and user mode components
  - Superior performance and scale
  - Well integrated with Distributed Transaction Services – supplies the state machine core for distributed transactions

# Distributed Transaction Flow



# KTM Features Include

- Full (fast) 2-phase commit on single machine
- Virtual clocks & save-points
- Roll-forward recovery
- Multiple log support
- Highly available (log failover)
- Integrated with protocol layers
- Kernel, Win32 and System.Transaction Interfaces

# Windows Logging Service

Dana Groff  
Program Manager  
Core File Services  
[dgroff@microsoft.com](mailto:dgroff@microsoft.com)

**PDC**<sup>03</sup>

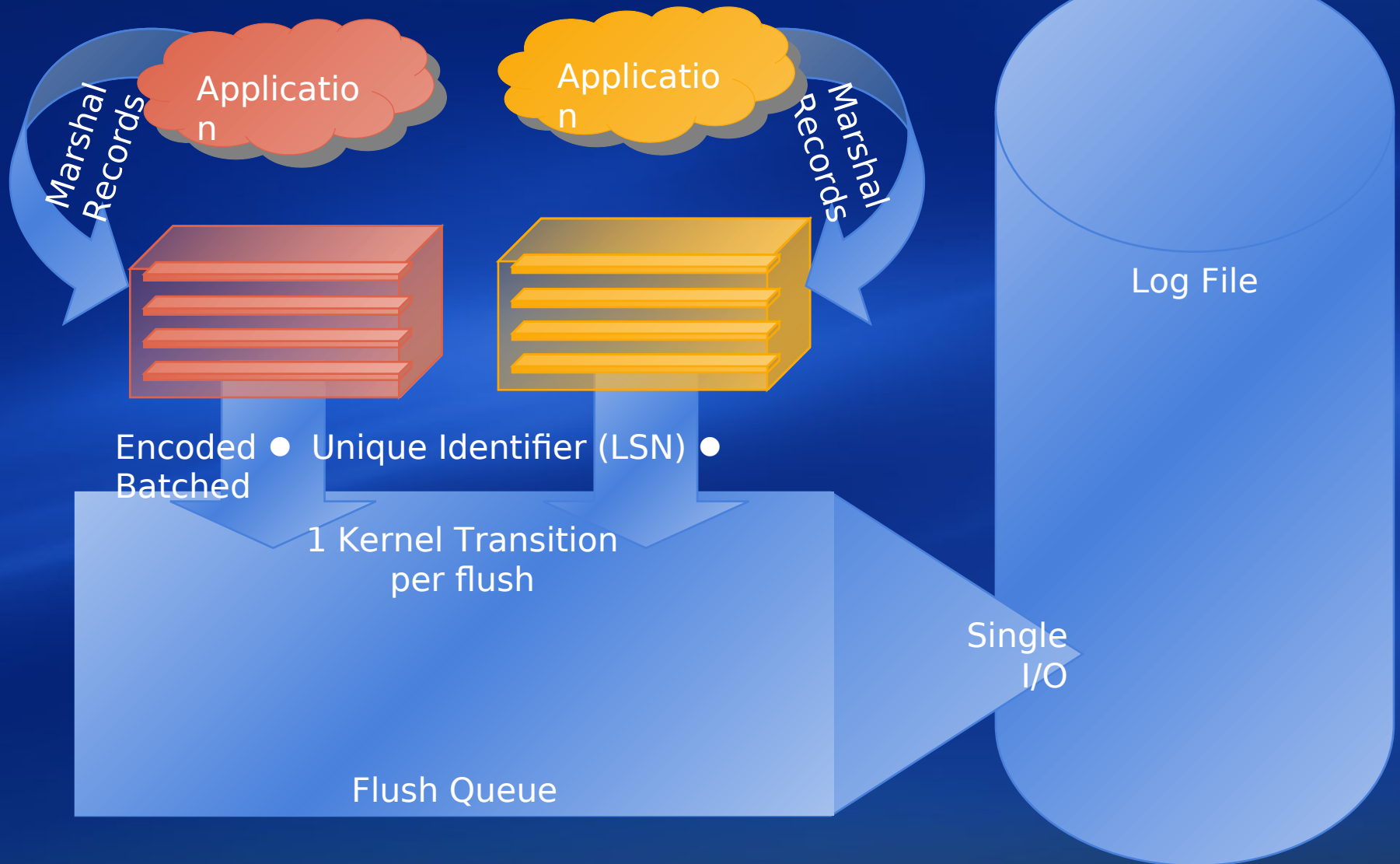
Make the connection



# Usage Scenarios

- When you need
  - Sequential Record Oriented Data
  - Grouped Records
  - Persistence
  - High performance
- Examples
  - Transactional Stores
  - Change Records for Replication
  - Event Logging

# Data Flow



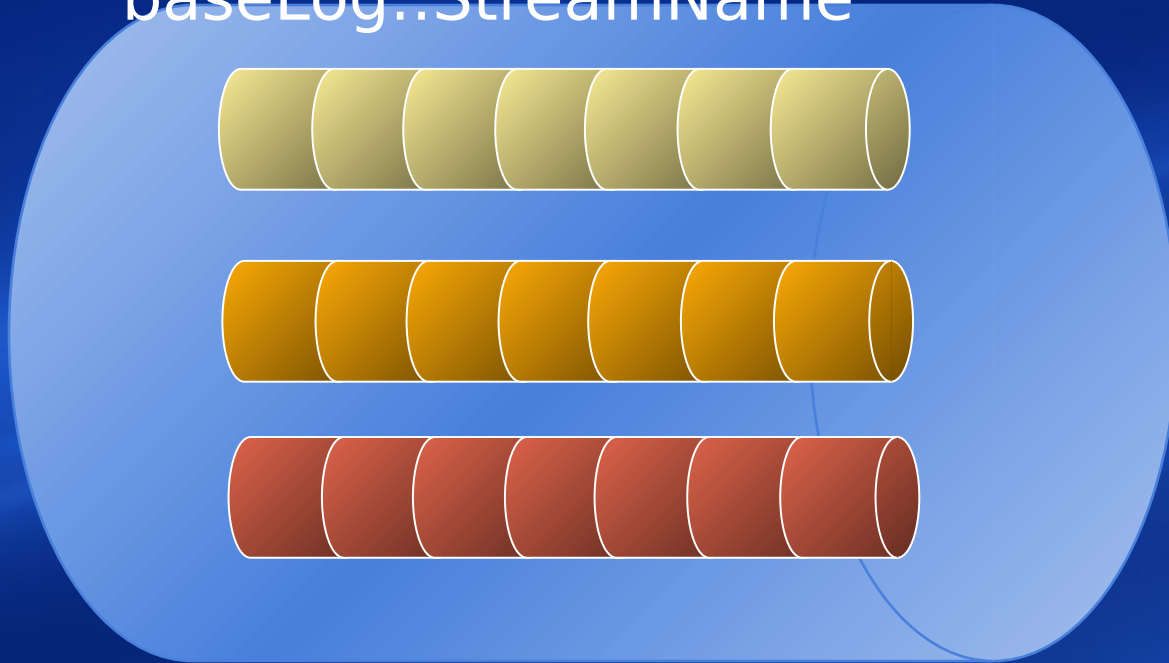
# Windows Logging Service (WinLS)

## demo

Create And Write To A Log

# Log Records in Streams

Log:/full/path/name/to/  
baseLog::StreamName



One or many streams in a log  
No need to change to code, just  
change the logname!

# Windows Logging Service (WinLS)

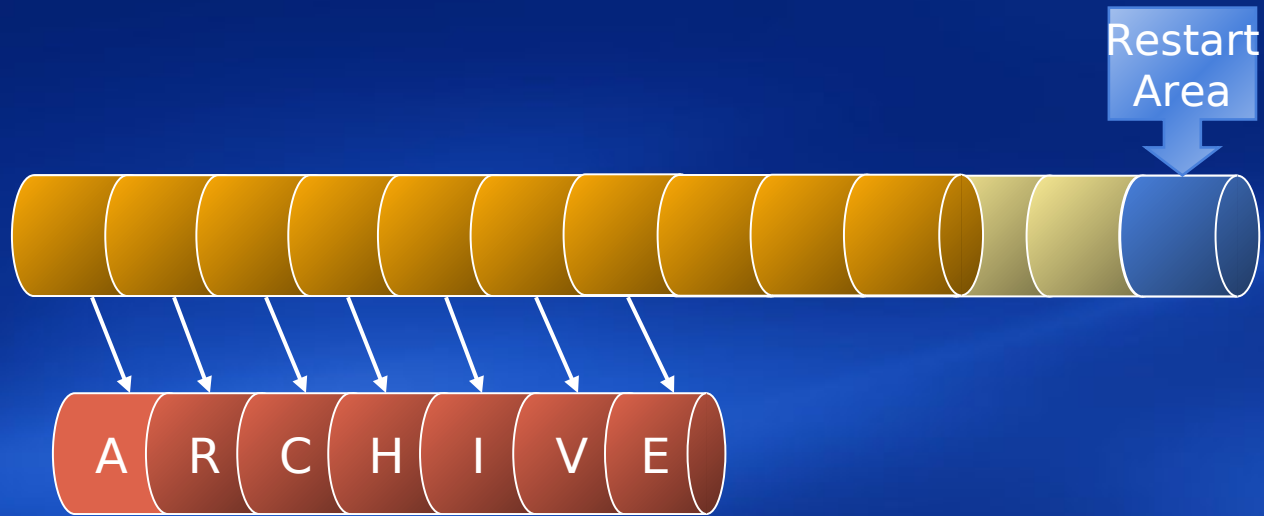
## demo

Easy Streams

**PDC**<sup>03</sup>

Make the connection

# Recycling Space



- Archiving without interrupting reading and writing to the log
- Restart Area : dedicated area to write restart / recovery data.



# Windows Logging Service (WinLS)

## demo

### Recycling Log Space

**PDC**<sup>03</sup>

Make the connection

# Key Features To Remember

- Fast, dependable, and optimized I/O
- Multiple streams without added coding
- Services for transactional logging
- APIs:
  - Managed Code (not available for PDC)
  - Win32
  - Kernel

# Community Resources

## Get Your Questions Answered!

- HOL
  - HOL-404 Transactional File System
  - *Meet the team Thursday 9 am to 11 am*
- Send Us Email
- 309 Foyer
  - connect with PDC 2003 Architecture and Infrastructure Speakers and their product team members
- Newsgroups:
  - microsoft.public.windows
    - ➡ developer winfx fundamentals



# PDC<sup>03</sup>

Make the connection

**Microsoft Professional Developers Conference 2003**

October 26 - 30, 2003, Los Angeles, CA

**Microsoft®**